

Nom Prénom	
------------	--

**- Listes, tuples, dictionnaires, Entrées -****TP numéro 3**

Objectif du TP	Prise en main des listes, tuples, dictionnaires et entrées utilisateurs Création, Manipulation, modification
Pré requis	Cours
Ressources à disposition	EDI
Capacités / compétences visées	Maîtriser les commandes et déclaration élémentaires
Travail demandé	Lire l'énoncé, appliquer les consignes
Suivi du professeur	Suivi du déroulement du travail des étudiants Passage auprès de chaque étudiant plusieurs fois dans le TP. Aide personnalisée. Réponse aux questions
Durée du TP	1 heure
Bilan / observations	

Contenu

1. Listes	2
2. Tuples	4
3. Dictionnaires	4
4. Entrées utilisateur	6

1. Listes

1.1 Modifier une liste

Soit la liste :

```
annee = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 10, 11, 12]
```

1.1.1 Supprimez les trois derniers éléments un par un, dans un premier temps

1.1.2 Puis rajoutez les mois 'Octobre', 'Novembre', 'Décembre' à la fin

1.1.3 Supprimez les trois derniers éléments un par un, dans un premier temps

Soit la liste :

```
annee = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 10, 11, 12]
```

Remplacez les trois derniers éléments par le nom du mois correspondant

1.1.4 Pour aller plus loin : la liste 'en compréhension'

Soit la liste :

```
x = [1, 2, 3, 4, 3, 5, 3, 1, 3, 2]
```

Python étant un langage procédural mais aussi fonctionnel, on peut appliquer des fonctions à l'ensemble des éléments d'une liste, par exemple, incrémenter tous les éléments d'une liste :

```
>>> [y+1 for y in x]
```

```
[2, 3, 4, 5, 4, 6, 4, 2, 4, 3]
```

Il faut lire l'instruction ainsi :

```
+1      est la fonction qu'on va appliquer à l'élément y  
for y   à toutes les valeurs que va prendre y  
in x    en parcourant la liste x (on retrouve le foreach d'autres langages)
```

Le résultat du traitement est une liste qui peut donc servir à créer une nouvelle liste par exemple

```
>>> resultat = [y+1 for y in x]
```

2. Tuples

Etant donné que les mois d'une année ne changent pas, créez une constante qui permet de stocker les mois d'une année :

```
>>>moisDeLannee = ('Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin',  
'Juillet', 'Août', 'Septembre', 'Octobre', 'Novembre', 'Décembre')
```

2.1 Accès aux éléments d'un tuple

Affichez le 4^e mois de l'année (même fonctionnement que pour les listes)

2.2 Vérifier la présence d'un élément dans un tuple

Vérifiez si le mois de 'mars' est présent dans grâce à l'instruction `in` dont le format est le suivant :
objetCherché `in` objectContenant, le résultat étant un booléen.

Faites ensuite la même chose pour le mois de 'Mars'

(Cela fonctionne aussi pour les listes)

3. Dictionnaires

Soit le dictionnaire suivant qui vous permet de connaître l'âge des membres d'un club :

```
age = {"pierre" : 35 , "paul" : 32 , "Jacques" : 27 , "andre" : 23}
```

3.1 Ajoutez des éléments au dictionnaire

De nouveaux membres rejoignent le club, ajoutez-les au dictionnaire :

david, 22 ans, veronique, 21 ans, sylvie, 30 ans, damien, 37 ans

M1 ESI

3.2 Accéder à une valeur à partir d'une clé

On vous demande quel est l'âge de sylvie, saisir la commande qui permet de répondre :

3.3 Accéder à une valeur à partir d'une clé

On vous demande si jean fait partie du club, saisir la commande qui permet de répondre (voir 'présence d'un élément dans un tuple')

3.4 Gérer des valeurs multiples

La base de données d'un club de sport utilise un dictionnaire nommé `club` dont les clés sont le nom des sportifs. Les valeurs contenant leur année de naissance, taille et poids sont, elles, stockées dans un tuple.

pierre durand, 1986, 1.72, 70 soit `['pierre durand']=(1986,1.72,70)`

victor dupont, 1987, 1.89, 57

paul dupuis, 1989, 1.60, 92

jean rieux, 1985, 1.88, 77

Créer le dictionnaire du club : `club = {}` et ajouter les autres sportifs

3.5 Afficher les données d'un sportif

Accédez aux données de 'paul dupuis' et initialisez les variables `dateNaissSportif`, `poidsSportif` et `tailleSportif` avec les valeurs du tuple correspondant.

Créez ensuite une chaîne de formatage `formatDonnees` qui permettra d'utiliser les variables pour afficher la chaîne suivante avec un `print()` :

Le sportif nommé Paul Dupuis est né en 1989, sa taille est de 1.6m et son poids est de 92Kg

Remarque : vous pouvez passer par une variable supplémentaire `varTuple` qui récupère la valeur à partir de la clé dans un premier temps puis initialiser les variables

4. Entrées utilisateur

4.1 Club sportif : variante

A partir de l'exercice précédent : Le nom du sportif devra être demandé à l'utilisateur avec `input` et sera stocké dans une variable `nomSportif`.

Il devra servir à faire les recherches dans la base de données afin d'alimenter les variables `dateNaissSportif`, `poidsSportif` et `tailleSportif`.

Dans la chaîne de formatage `formatDonnees`, vous devrez utiliser aussi la variable `nomSportif`